National University of Singapore

School of Computing

Semester 4 (2015/2016)

CS2010 - Data Structures and Algorithms II

Midterm Test (20%)

Tuesday, 12 July 2016, 10.05am-11.35am (90 minutes)

INSTRUCTIONS TO CANDIDATES:

1. Do **NOT** open this question paper until you are told to do so.
2. Midterm Test is conducted at COM1-0204 (SR2).
3. This question paper contains THREE (3) sections with sub-questions.
   It comprises EIGHT (8) printed pages, including this page.
4. Write all your answers in this question paper, **but only in the space provided**.
   You can use either pen or pencil. Just make sure that you write **legibly**!
   Important tips: Pace yourself! Do **not** spend too much time on one (hard) question.
5. This is an **Open Book Examination**. You can check the lecture notes, tutorial files, problem set files, Steven's 'Competitive Programming 1/2/2.5/3' book, or any other printed material that you think will be useful. But remember that the more time that you spend flipping through your files implies that you have less time to actually answering the questions.
6. **Please write your Matriculation Number in the box below.**
   But do not write your name in order to facilitate unbiased grading.

A	0	1						

7. All the best :).

---

1 Analysis (15 marks)

Prove (the statement is correct) or disprove (the statement is wrong) the statements below.

1. The depths of any two leaves in a Binary (Max or Min) Heap differ by at most 1.

2. Given a standard BST (not a self balancing BST) of N distinct integers, we delete an integer X that is guaranteed to exist in that BST and X is a leaf vertex. Then, we re-insert X again. The resulting structure of that BST will be exactly the same.

3. Insertion into an AVL Tree with N vertices can trigger up to $\log N$ rotations.

4. After doing `Find-Set(x)` operation in Union-Find Disjoint Sets data structure with the path compression heuristic activated, the real height of the tree that contains `x` (not the rank of `x`) **always decreases**. Additional constraint: `x` is not the root nor child of the root of the tree.

5. Given a connected graph with V vertices and $E = V - 1$ edges, that graph must be **planar**.
   Definition: A planar graph is a graph that can be embedded in the plane (2-D surface).
   In other words, it can be drawn in such a way that no edges cross each other.

2 Not Yet in VisuAlgo Online Quiz (25 marks)

2.1 Create Binary (Min) Heap with K Inversion(s) (15 marks)

New Definition: Given an array A, if $i < j$ and $A_i > A_j$, then the pair (i, j) is called an inversion of A and requires one swap between A_i and A_j to make $A_i \leq A_j$. An array A is said to have K inversions if it requires at minimum K swaps to make array A sorted in (ascending) order. For example, $A = [3, 2, 1]$ requires $K = 3$ swaps to make it sorted to $A' = [1, 2, 3]$.

Revision: Binary **Min** Heap that contains N integers can be represented in a compact array $A = [A_1, A_2, \ldots, A_{N-1}, A_N]$ and the smallest integer must be in A_1.

Someone, not Steven, claims that: "The compact array A of a Binary (Min) Heap looks like a 'nearly sorted' array as the smaller integers are at the front of the array".

Is that statement true? Let's investigate.

Given $N = 15$ distinct integers inside the compact array A, for simplicity, assume $A = [1, 2, ..., 14, 15]$, give a permutation of that $N = 15$ distinct integers of A so that A has:

-. (no mark). $K = 0$ inversion (**just for illustration**).
My answer: A = [**1,2,...,14,15**], My explanation: **Those 15 distinct integers are already sorted in ascending order (so $K = 0$ inversion) and also satisfy Binary Min Heap property.**
a (2 marks). $K = 1$ inversion.
My answer: A = [_____], My explanation:

b (3 marks). $K = 2$ inversions.
My answer: A = [_____], My explanation:

c (4 marks). $K \geq 35$ inversions :O.
My answer: A = [_____], My explanation:

d (6 marks). $K = 53$ inversions, the maximum possible answer for $N = 15$:O:O.
My answer: A = [_____], My explanation:

2.2 Create Binary Search Tree (BST) of Height H (10 marks)

Old Definition: The height H of a tree is **the number of edges** from the root to its deepest leaf.

Given $N = 15$ distinct integers, for simplicity, assume they are [1,2,...,14,15], give a permutation of these N distinct integers so that if we insert them one by one into an initially empty standard BST (not a self-balancing BST) according to that permutation, the height of the resulting BST is:

a (2 marks). $H = 14$.
My answer: _____, My explanation:

b (4 marks). $H = 3$.
My answer: _____, My explanation:

c (4 marks). $H = 10$.
My answer: _____, My explanation:

3 Not Suitable for VisuAlgo Online Quiz (60 marks)

3.1 Binary Indexing (15 marks)

Definition: A **perfect** binary tree has a distinguished vertex called the root which is usually drawn at the top. Each vertex has **exactly two children** except the vertices in the lowest layer, which we call the leaves.

For this question, we label the vertices of a perfect binary tree with integers as follows: We start at the bottom right leaf which gets integer 1 and then label vertices on the same level in increasing order **from right to left**. After finishing a level, we move to the rightmost vertex in the **level above** and label all the vertices on that level **from right to left**. We proceed in this fashion until we reach the root, which will get the last integer N, where N is the number of vertices in the tree.

When we describe a vertex in the tree, we mention a path starting at the root and going down toward the leaves. At each non-leaf vertex we can either go left ('L') or right ('R').

Your task is this, given an integer H, denoting the height of this perfect binary tree, and the a string S, denoting a path in the tree starting from the root, calculate the label of the described vertex, i.e. write a function `label(H, S)`. In Figure 1, we show two examples: `label(3, ''LR'') = 11` and `label(3, ''RRL'') = 2`.

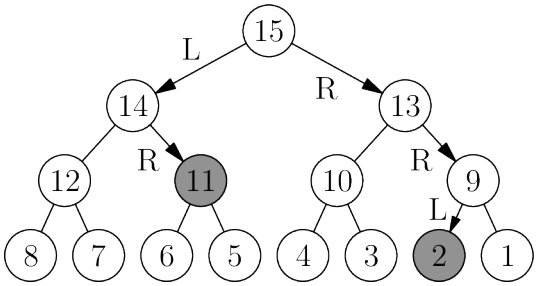

Figure 1: Example of a labeled perfect binary tree of height $H = 3$ with two marked paths from the root. Path "LR" leads to label 11 while path "RRL" leads to label 2. The root has label 15.

Constraints: $1 \leq H \leq 30$; S consists of the letters 'L' and 'R' only. The letter 'L'/'R' denotes choosing the left/right child, respectively. S **may be empty** and is **at most** H **letters**.

For this question, please write your answer in Java below.

```
public static int label(int H, String S) {
```

```

```

}

Problem credit: Lukáš Poláček, `https://open.kattis.com/problems/numbertree`

3.2 Cleaning Duplicates (15 marks)

Given an array A containing N integers, possibly containing duplicates, we want to clean A so that if $A_i = A_j$ where $i < j$, we keep A_i as it is but set $A_j = -1$. For example, if $A = \{10, 7, 3, 2, 10, 8, 10, 7\}$, then after cleaning, we will have $A' = \{10, 7, 3, 2, -1, 8, -1, -1\}$

Design the best algorithm to perform this cleaning task and **analyze its worst time complexity using Big O notation**. You are free to use any data structure that you see fit to do this task. For this question, you can answer in Java or in Pseudocode.

3.3 Median Filter (15 marks)

An N-integers array A of noisy data can often be smoothed out using a median filter. To do this we take a window of length K (K is guaranteed to be odd and $K \leq N$) integers and slide it over our sequence, at each location outputting the median integer within the window.

The first window covers integers from index 0 to index $K - 1$ and the smoothed data will contain $N - K + 1$ integers.

For example, given $A = \{70, 3, 5, 8, 6, 35, 7\}$ with $N = 7$ and $K = 3$, applying such median filter will produce the following result $A' = \{5, 5, 6, 8, 7\}$ with $7 - 3 + 1 = 5$ integers because:

- The median of (70,3,5) = 5,
- The median of (3,5,8) = 5,
- The median of (5,8,6) = 6,
- The median of (8,6,35) = 8, and finally
- The median of (6,35,7) = 7.

Design the best algorithm to perform such median filter and **analyze its worst time complexity using Big O notation**. You are free to use any data structure that you see fit to do this task. For this question, you can answer in Java or in Pseudocode.

3.4 Graph Data Structures (15 marks)

Definition 1: A star graph S_M is the complete bipartite graph $K_{1,M}$. Notice that it has $M + 1$ vertices. Basically it has one vertex (which can be any of the $M + 1$ vertices) that is connected to every other vertex (so M edges) and there is no other edge in that star graph.

Definition 2: The cycle graph on N vertices is called the N-cycle and usually denoted C_N. There are N edges in a cycle graph C_N.

Definition 3: The wheel graph W_N is a graph on N vertices constructed by connecting a single vertex (which can be any of the N vertices) to every vertex in the other $(N-1)$-cycle (defined in the cycle graph above). Thus we have $(N-1)+(N-1) = 2N-2$ edges.

Three examples are shown in Figure 2.

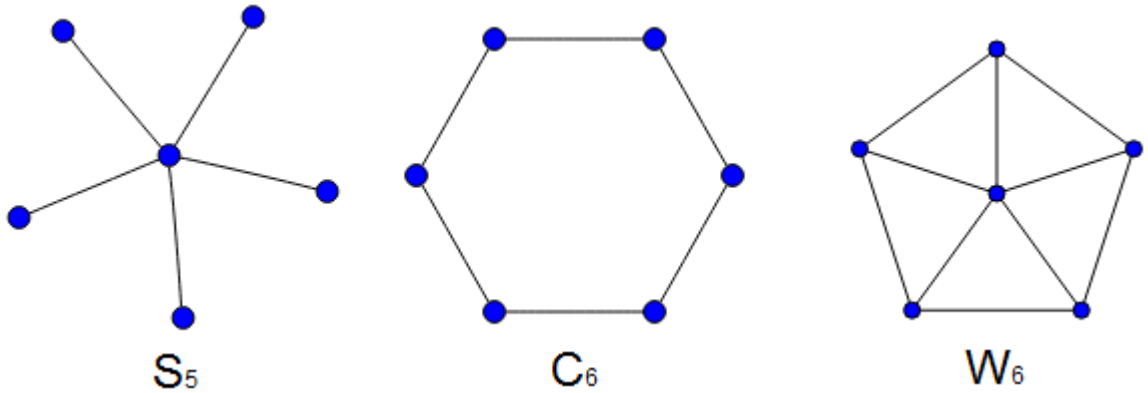

Figure 2: Example of S_5, C_6, and W_6.

What are the best graph data structures to store a star graph S_M (that has $M + 1$ vertices), a cycle graph C_N, and a wheel graph G_N (5 marks for each graph type)? You are allowed to permute vertex labels before storing the graph. All graphs are unweighted and undirected. Is it Adjacency Matrix?, Adjacency List?, Edge List?, or something else not covered in the class? Please **analyze, compare, and defend** your answers.

The best graph data structure to store S_M is:

The best graph data structure to store C_N is:

The best graph data structure to store W_N is:

BONUS QUESTION (1 mark): During the introductory lecture on Tuesday, 21 June 2016, Steven showed a fictional comic character that appeared briefly during the slide titled 'Typical Class Profile (CS2010 in S4)'. What is the name of that character?

– End of this Paper –

Section	Maximum Marks	Your Marks	Comments from Grader
1	15		
2	25	_____ + _____ = _____	
3	60	_____ + _____ + _____ + _____ = _____	
Bonus	1		
Total	Min(100, Your Marks)		